# Default Data Source Configuration

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

# First Data Source (Primary)

spring.datasource.primary.url=jdbc:mysql://localhost:3306/primarydb

spring.datasource.primary.username=root

spring.datasource.primary.password=rootpassword

spring.datasource.primary.driver-class-name=com.mysql.cj.jdbc.Driver

# Second Data Source (Secondary)

spring.datasource.secondary.url=jdbc:mysql://localhost:3306/secondarydb

spring.datasource.secondary.username=root

spring.datasource.secondary.password=secondarypassword

spring.datasource.secondary.driver-class-name=com.mysql.cj.jdbc.Driver

package com.example.employeemanagementsystem.config;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.boot.orm.jpa.EntityManagerFactoryBuilder;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import javax.sql.DataSource;

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.employeemanagementsystem.primary.repository",

entityManagerFactoryRef = "primaryEntityManagerFactory",

transactionManagerRef = "primaryTransactionManager"

)

public class PrimaryDataSourceConfig {

@Bean(name = "primaryDataSource")

public DataSource primaryDataSource() {

return DataSourceBuilder.create()

.driverClassName("com.mysql.cj.jdbc.Driver")

.url("jdbc:mysql://localhost:3306/primarydb")

.username("root")

.password("rootpassword")

.build();

}

@Bean(name = "primaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory(

EntityManagerFactoryBuilder builder,

@Qualifier("primaryDataSource") DataSource primaryDataSource) {

return builder

.dataSource(primaryDataSource)

.packages("com.example.employeemanagementsystem.primary.entity")

.persistenceUnit("primary")

.build();

}

@Bean(name = "primaryTransactionManager")

public PlatformTransactionManager primaryTransactionManager(

@Qualifier("primaryEntityManagerFactory") LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory) {

return new JpaTransactionManager(primaryEntityManagerFactory.getObject());

}

}

package com.example.employeemanagementsystem.config;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.boot.orm.jpa.EntityManagerFactoryBuilder;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import javax.sql.DataSource;

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.employeemanagementsystem.secondary.repository",

entityManagerFactoryRef = "secondaryEntityManagerFactory",

transactionManagerRef = "secondaryTransactionManager"

)

public class SecondaryDataSourceConfig {

@Bean(name = "secondaryDataSource")

public DataSource secondaryDataSource() {

return DataSourceBuilder.create()

.driverClassName("com.mysql.cj.jdbc.Driver")

.url("jdbc:mysql://localhost:3306/secondarydb")

.username("root")

.password("secondarypassword")

.build();

}

@Bean(name = "secondaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean secondaryEntityManagerFactory(

EntityManagerFactoryBuilder builder,

@Qualifier("secondaryDataSource") DataSource secondaryDataSource) {

return builder

.dataSource(secondaryDataSource)

.packages("com.example.employeemanagementsystem.secondary.entity")

.persistenceUnit("secondary")

.build();

}

@Bean(name = "secondaryTransactionManager")

public PlatformTransactionManager secondaryTransactionManager(

@Qualifier("secondaryEntityManagerFactory") LocalContainerEntityManagerFactoryBean secondaryEntityManagerFactory) {

return new JpaTransactionManager(secondaryEntityManagerFactory.getObject());

}

}

package com.example.employeemanagementsystem.primary.repository;

import com.example.employeemanagementsystem.primary.entity.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Primary data source repository methods

}

package com.example.employeemanagementsystem.secondary.repository;

import com.example.employeemanagementsystem.secondary.entity.Department;

import org.springframework.data.jpa.repository.JpaRepository;

public interface DepartmentRepository extends JpaRepository<Department, Long> {

// Secondary data source repository methods

}